TITLE: A Multiobjective Approach to Transonic Wing Design by Means of Genetic Algorithms

TITLE: Aerodynamic Design and Optimisation of Flight Vehicles in a Concurrent Multi-Disciplinary Environment [la Conception et l'optimisation aerodynamiques des vehicules aeriens dans un environnement pluridisciplinaire et simultane]

A multiobjective approach to transonic wing design by means of genetic algorithms

A. Vicini, D. Quagliarella
CIRA – Centro Italiano Ricerche Aerospaziali
via Maiorise 81043 Capua (CE), Italy

1. Summary

In this work a transonic wing design problem is faced by means of a multiobjective genetic algorithm, and using a full potential flow model. The applications here presented regard both planform and wing section optimization. It is shown how both geometric and aerodynamic constraints can be taken into account, and how the multiobjective approach to optimization can be an effective way to handle conflicting design criteria. An interpolation technique allowing a better approximation of Pareto fronts is described. Two possible ways of improving the computational efficiency of the genetic algorithm, namely a parallel implementation of the code and a hybrid optimization approach, are presented.

2. Introduction

Transonic wing design is a very complex task, even if considered from a purely aerodynamic point of view. In fact, several criteria must be met in the design of any efficient transport wing, including good drag characteristics, buffet boundary high enough to permit cruising at design lift coefficients, no pitch-up tendencies near stall, no unsatisfactory off-design performances etc.[1]

Recent advances in computational techniques have made it possible to effectively address much more complex design problems than was previously possible, and concurrently reduce the design cycle flow time. The rapid improvements in the speed of computers have then originated a growing development of numerical optimization techniques for applications to aerodynamic design. Several techniques are today available[2], from mature gradient based methods to more recent approaches like automatic differentiation, control theory based methods and genetic algorithms (GAs). It is not possible, generally speaking, to state the overall superiority of one method over the others; in fact, there are several characteristics that must be considered, and that may assume different importance depending on the specific problem at hand. Among these characteristics we may mention: 1) the generality of the formulation, i.e. the possibility to rapidly set up different optimization problems, including also the use of different analysis tools; 2) robustness, intended as the capability to find global optima and reduce the need of human interaction and expertise; 3) the possibility to deal with multiple design objectives and constraints; 4) computational efficiency, for a practical use of the design approach.

On the other hand, because aerodynamic shape design represents only a part of the overall design of a flying vehicle, and because the need for an effective multidisciplinary approach to the design task is rising, it is important to identify the difficulties which are typical of multidisciplinary environments. Among these we may mention the necessity to operate at system level, and consequently managing and interrelating design objectives of different nature. Moreover, the dimensionality of the design space may increase to a point where traditional mathematical programming methods are likely to find severe difficulties. The design problem may be characterized by a mix of continuous, discrete and integer design variables, and the resulting design space can be non convex or even disjointed. For all these reasons, optimization methods which do not rely on the computation of gradients, in particular evolutionary programming and genetic algorithms, are receiving a considerable growth of interest[3]; in fact, these strategies are less susceptible to pitfalls of convergence to local optima, and generally offer a more robust approach to complex design problems. Indeed, the major weakness of such methods lies in their poor computational efficiency, which still prevents their practical use when the evaluation of the cost function is expensive, as happens with three-dimensional aerodynamic problems and complex flow models.

In this work a multiobjective genetic algorithm[4] is used for a transonic wing design problem, using a full potential flow solver as aerodynamic analysis tool. In the design examples here presented, the optimization of the wing planform is first addressed through a multiobjective approach, by taking into account both aerodynamic and structural weight considerations; afterwards, the wing sections are optimized to improve the aerodynamic performances by decreasing drag for a specified lift, while controlling some geometric characteristics of the airfoils and the pitching moment coefficient. Two different approaches for the enhancement of the computational efficiency of the procedure are illustrated; the first one is a parallel implementation of the code, which exploits the favourable structure of the genetic algorithm, while the second one is a hybrid approach obtained by including a gradient based optimization routine among the set of operators of the GA.

Paper presented at the RTO AVT Symposium on "Aerodynamic Design and Optimisation of Flight Vehicles in a Concurrent Multi-Disciplinary Environment", held in Ottawa, Canada, 18-21 October 1999, and published in RTO MP-35.

3. The Genetic Algorithm

Genetic algorithms belong to the class of evolutionary strategies, which common feature is the attempt to emulate the mechanisms of biological evolution. In their original formulation[5], a set of possible solutions to the problem at hand (population) are coded into bit strings (*chromosomes*); a number of operators are then used for the transformation and improvement of these solutions by evolving through subsequent generations[6]. The first component of a GA is therefore a scheme that allows for a coded representation of possible solutions; as stated above, a bit string codification is usually adopted, by representing each design variable through a fixed length binary number, and linking together all the coded variables in a single string. Then, there must be a criteria for the evaluation of the *fitness* of each individual of the population, which is a measure of how good the corresponding solutions are, allowing for a ranking of the individuals of the population; this criteria is of course problem dependent. The functions used for the simulation of the biological evolution when applied to the chromosome strings must then be devised. A criteria for the selection of the pairs of individuals that are going to reproduce must be chosen, such that selection probability is higher for individuals characterized by higher fitness. The parents chosen for reproduction are then mated through a *crossover* operator, which allows the recombination of their chromosomes; finally, the two strings obtained can undergo a mutation, consisting in a random variation of a little portion of the information coded in them. A number of different selection, crossover and mutation operators can be used; though the choice of these operators may have strong influence on the performances of the procedure, for a given optimization problem[7], it doesn't affect the basic scheme of the algorithm.

As already stated, a peculiar feature offered by GAs is their capability to face multiobjective optimization. When several design goals need to be achieved in an optimization problem, these are usually combined together so that a single scalar objective function is obtained. In this way, the problem becomes amenable to all classical optimization algorithms. The drawback of this approach is that the solution of the problem is strongly dependent on the (arbitrary) choice of the relative weights assigned to the objectives; moreover, if the objectives to be minimized are of different nature, as happens for example when multidisciplinary optimization problems are faced, it is difficult to understand how to interrelate them properly.

It can be convenient to follow a different approach, by classifying all potential solutions to the multiobjective optimization problem into *dominated* and *non dominated* (Pareto optimal) solutions. One solution is not dominated if there is no other solution which is better with respect to all design objectives. The Pareto front is the set of all the non dominated solutions; it follows from the definition that, if a solution belongs to the Pareto Front, it is not possible to improve one of the objectives without deteriorating some of the others.

By virtue of their structure, GAs are capable of facing multiobjective design problems in a more direct way; in fact, by selecting individuals according to the domination criteria instead of on the basis of a single fitness value, the set of Pareto optimal solutions can be closely approximated. In this way, a number of possible alternative solutions are obtained, each one meeting the requirements of the problem at different levels of compromise. Hence, the characterizing feature of a multiobjective GA is the introduction of the domination criteria in the method used for individuals selection. In this work, this is accomplished through a *random-walk* operator: the current population is distributed over a toroidal landscape, a starting point is chosen at random, and the parents are selected as the locally non dominated individuals met in two subsequent *walks*, of a given number of steps, from that starting point; if more non dominated individuals are met, the first one encountered is selected.

In some of the applications here illustrated, an interpolation technique has been used to improve the quality Pareto front; this technique consists in what follows:

1) after each new generation G_i is completed, it is merged with the current Pareto front (P_{i-1}), and the new set of non dominated individuals (updated Pareto front, $P_i^{'}$) is extracted and stored;

2) afterwards, all the couples of adjacent individuals along the front are considered, and, if their distance (measured in the objectives plane) is higher than a specified percentage of the average distance between adjacent elements of the Pareto front, a new individual is generated by linear interpolation of their design variables;

3) after their evaluation, all the new individuals thus generated are merged with the current Pareto front set $P_i^{'}$, and the set of non dominated individuals P_i is extracted.

A flow chart of the resulting algorithm is illustrated in Fig. 1. This procedure has two positive effects: the first one is the enrichment of the Pareto front at the end of each generation, as many of the elements obtained through interpolation belong to the non dominated set; the second advantage is the exploitation of the first one, and it is obtained when an elitist strategy is adopted by selecting some of the individuals in the reproduction phase from the current Pareto front. The combination of these two effects allows to improve the approximation of the Pareto front with more and better distributed individuals.

4. Hybrid GA

Coupling a genetic algorithm with a different optimization technique can be an effective way to overcome its lack of efficiency while preserving its favourable features. Of course, many different strategies to hybridize the GA can be realized; the ideal approach is to combine the best features of both methods, so as to provide results better than those obtainable using either of the two techniques.

A simple GA may by itself be considered as the combination of two different search techniques, namely crossover and mutation, that are characterized by different behaviours. Crossover is a powerful tool to search the design space and single out the region where the global optima lie, but it lacks the capability of effectively refine the sub-optimal solutions found. On the other hand, mutation has a more local effect, since the modifications it produces are generally smaller in the coded parameter space. Hence, mutation has two important roles in simple GAs, i.e. to

provide the capability to effectively refine sub-optimal solutions, and to re-introduce in the population the alleles lost by the repeated application of crossover, maintaining population diversity. However, there is a broad class of problems, namely the ones where the fitness function is differentiable, for which gradient based techniques are much more efficient to locally improve a given solution. This suggests the introduction of a gradient based routine among the set of operators of the GA; mutation is then prevalently left with the role of keeping the diversity among population elements at an optimum level.

The genetic algorithm developed adopts a bit string codification of the design variables; anyway, this does not prevent the use of operators requiring real number list encoding, such as extended intermediate crossover and word level mutation.[4] In these cases the binary string is decoded into a real number list, the operator is applied and the set of modified variables is encoded back into a bit string. This scheme allows the use of a free mix of different type of operators.

A routine performing a gradient based optimization (with a conjugate gradients technique) has then been included as a GA operator, and called "hill climbing operator" (HCO). This operator is used as follows: through the application of the selection, crossover and mutation operators, an intermediate generation is created from the current one; afterwards, if the hybrid option is activated, some individuals may be selected and fed into the hill climbing operator to be improved, and then introduced into the new generation, as sketched in Fig. 1. Regarding the choice of the elements to be fed into the gradient operator, in the case of single objective optimization three different strategies are possible:

1. only the best fit individual of the current generation is chosen;
2. a number of elements determined by an assigned probability is picked using the selection operator;
3. a number of elements determined by an assigned probability is picked in a purely random fashion.

Of course, these strategies are characterized different levels of selection pressure, decreasing from strategy #1 to strategy #3; the relative performance will therefore depend on the optimization problem. The above described scheme can be naturally extended for multiobjective optimization: in this case, strategy #1 becomes the (random) selection of a number of elements determined by an assigned probability from the current set of Pareto optimal solutions, while strategies #2 and #3 remain the same. Of course, the hill climbing operator is by its nature capable of dealing only with scalar objective functions; thus, when multiobjective problems are faced, the objective function fed into the HCO is obtained through a weighted linear combination of the n problem objectives, i.e. as $obj = \alpha\ obj_1 + (1-\alpha)\ obj_2$ in the case of $n = 2$. The weighting factor α can be chosen at random or assigned explicitly to favour one of the objectives.

An airfoil inverse design problem is used to illustrate the behaviour of the hybrid procedure. In this case, the objective function to be minimized, for a given Mach number and angle of attack, is computed by:

$$obj = 10 \int_S \left(c_p - c_p^{(t)}\right)^2 ds \qquad (1)$$

where c_p and $c_p^{(t)}$ are the current and target pressure distributions, respectively, and S is the current airfoil contour; the fitness is then obtained as $f = 1/obj^2$. A full potential transonic flow solver, with non-conservative formulation, has been used to calculate the flow field. The airfoil geometry is represented by means of two 5^{th} order B-spline curves, for the upper and lower parts. The coordinates of the control points of the B-spline constitute the design variables;[7] 7 control points are used both for the upper and lower surfaces of the airfoil, including those fixed at the leading and trailing edges, for a total of 18 design variables (the first control points at the leading edge can move only in direction y). The problem here presented consists in the reconstruction of the CAST-10 airfoil[8] at $M = 0.765$, $\alpha = 0$. This problem has been solved using a NACA 0012 as initial guess, which can be considered an absolutely generic starting point.

The design variables have been encoded using 8 bit strings (giving a chromosome length of 144 bits), and a 50 individuals population evolved for 100 generations.A 3-step random walk was used for individuals selection, with one-point crossover ($p_c = 1$) and word level mutation ($p_m = 0.02$). The hybrid strategies have been activated so as to select on average only one individual every other generation, and carry out 2 gradient iterations $N_{it} = 2$, $\eta = 1$) Hence, to consider the same total number of objective function evaluations, the hybrid strategies must be judged approximately at generation 70. Fig. 3 illustrates the convergence histories, each one averaged over 10 successive trials characterized by different starting populations, obtained with the GA and with the corresponding hybrid strategies. The convergence history obtained by the application of the gradient based method by itself is also shown in the same figure; in this case, of course, there are no generations of individuals, but the convergence history is reported in such a way that the number of required objective function evaluations can be obtained from the same scale (1 generation = 50 evaluations). Besides, it must be noted that a restart procedure had to be used in this case to take the solution out of a local minimum where it got stuck after a few iterations.

As can be seen, for a given GA, hybridization is always beneficial, meaning that a better result can be found with the same amount of computations, or that the same result can be obtained with a substantial reduction of computation needed (ranging in this case from 50 to 75%). In particular, strategy #1, when the hill climbing operator is applied only to the best fit individuals, appear as the less effective, probably due to an excessive selection pressure. At the same time, the behaviour of the gradient based method is considerably improved from the point of view of the robustness.

Another important characteristic that needs to be considered is the statistical dispersion of the results obtained starting from different initial populations; in fact, if it is correct to judge the convergence characteristics of a given GA by averaging the results of a number of runs, from an application-oriented point of view it is more important for the algorithm to guarantee satisfactory convergence performances even on a single run basis. Fig. 4 shows all the values of the objective function obtained at the end of each of the 10 different runs, for each one of the algorithms used; it can be observed how the scatter of the results provided by the basic GA is much higher than that obtained using

the corresponding hybrid algorithms.

5. Parallel GA

The structure of GAs naturally adapts to the use of parallel computing, allowing to obtain remarkable efficiency improvements. When a population of individuals needs to be evaluated, it can be divided into a number of subpopulations which are sent to different processors; in this way, it is possible to evaluate the whole population in the same time required by a single analysis, if a sufficient number of processors is available. It is then also possible to take advantage by the splitting of the population, by using particular techniques. It has been shown for example[9] that, for some optimization problems, it can be convenient to let the different subpopulations evolve separately, with migrations between subpopulations allowed only to a controlled degree, and following particular strategies. Besides, it can be noted that, although this approach can be easily and naturally implemented on parallel computers, it is not exclusive domain of such kind of machines.

In this paper only a description of the parallel GA will be given. The individuals of each new generation are evaluated in a single loop, that follows the recombination phase and that can be carried out in parallel. The parallel programming model adopted relies on shared memory multiprocessing and the parallelism is implemented at the process level. The parallel machine adopted is a SGI POWER CHALLENGE system with 16 R-10000 processors. The parallel code has been implemented using the lightweight UNIX process primitives available on this machine[10].

The software is organized following the master-slave paradigm. In the initialization phase that precedes the first execution of the evaluation loop, the master process creates a pool containing a number of processes equal to the maximum number of processors available for the computation (NPROC). The child processes created are immediately put in a wait state, in which they remain until they receive a "go ahead" signal from the master to start the computation. This architecture has been chosen to avoid the inefficiency of creating a child process every time a computation is needed and killing it at the end.

When the master process enters the evaluation loop, it splits the population in subsets of maximum NPROC elements and then, for each subset, it copies the data relevant to a single computation in a shared memory area that is accessed by one of the child processes; afterwards, a signal is sent to the child, through a standard POSIX semaphore, so that the computation can begin. When the child terminates its computation, it copies the results in a memory region accessible to the master, but not over-writable by the other children. Thereafter, it sends a completion signal to the master (using another POSIX semaphore). The master waits for the completion of all child processes in a synchronization point. When this happens, the master collects the results and starts with the next subset of population elements. The child processes are terminated at the very end of the program, when all the evaluation loops related to each generation have been completed.

It can be observed that this architecture is efficient only when each sub-process has an even computational charge. If this does not apply, the computation process can loose efficiency because the master has to wait at the synchronization point for the completion of the slowest process. Fortunately, this is not the case for the kind of application considered so that the efficiency decay can be neglected. However it is not difficult to modify the master so that it can assign a new task to a child as soon as this has completed the previous one.

6. Applications to wing design

6.1. Wing geometry parametrization

The parametrization of the wing geometry adopted for this work includes the possibilities to modify both the wing planform and the shape of the sections. For the sake of simplicity, in the application that will be illustrated the wing planform has been kept trapezoidal, so that all geometric characteristics vary linearly from the root section to the tip. In this case, a total of 6 design variables may be used: 4 of these act directly on the wing planform, namely the taper ratio λ, the sweep angle at 25% of the chord Λ, the aspect ratio AR and the twist angle θ; moreover, the (percent) thickness at the wing root and tip have also been included among the design parameters. The geometry modifications are made in such a way that the wing surface is kept constant, so that the average wing loading is not changed during optimization.

The wing geometry is assigned through a number of spanwise "control" sections, which shape can be modified in the optimization loop; intermediate sections are obtained by linear interpolation. The airfoil shape for each control section has been represented as a linear combination of the initial shape and a prescribed number of modification functions; the characteristic of these modification functions is that they are obtained as the difference between the initial airfoil shape and the shape of existing airfoils chosen from an available database, as follows:

$$y(x) = y_o(x) + \sum_{i=1}^{N} w_i \left[y_i(x) - y_o(x) \right] \qquad (2)$$

where y_o is the initial airfoil shape, y_i is the geometry of the $i-th$ airfoil of the database and w_i is the design variable associated with it. In this way, the design variable associated with each of these modification functions has an effect on the whole airfoil shape.

6.2. Wing planform design

As anticipated, the design of a transonic wing planform will first be illustrated; the design has been accomplished by minimizing inviscid aerodynamic drag, which combines induced and wave drag, and structural weight, at a given Mach number $M = 0.84$ and lift coefficient $c_L = 0.3$. The starting geometry chosen is the ONERA M6 wing[11]; this wing has an aspect ratio AR=3.8, a leading edge sweep $\Lambda = 30°$, a taper ratio $\lambda = 0.562$, and is untwisted. The shape of the airfoil is symmetrical, with a maximum thickness of about 9.8 % chord. A total of 5 design variables have been used; in Table 1 the initial values of the design parameters are reported together with the prescribed allowable ranges. The thickness at the wing tip has been fixed at the original value $t/c\,|_t$= 9.8 %. For all the cases that will be described, the constraint on the lift coefficient has been satisfied by

using the angle of attack as a free parameter, and letting the flow solver adjust it to meet the desired lift value.

The wing twist is distributed symmetrically between the root and the tip, so that a twist angle θ corresponds to an increase of local incidence of $\theta/2$ at the tip, and a decrease of $\theta/2$ at the root. The wing weight is computed using the algebraic equation of Ref. 12; this equation combines analytical and empirical (statistical) methods, and shows design sensitivity and prediction accuracy that make it possible to use it with success for preliminary design.

As can be seen from Table 2, individuals selection has been carried out through a 2 step random walk, with one-point crossover ($p_c = 1$) and bit mutation ($p_m = 0.1$), and a population of 16 individuals was let evolve for 10 generations.

The Pareto front obtained using the Pareto interpolation technique previously described is illustrated in Fig. 5 (where W_o and c_{Do} are the values of the original M6 wing), together with the planform of a few wings corresponding to the indicated positions along the front; the front is populated by 123 individuals. In the same picture also the Pareto front computed without the described interpolation technique is illustrated. In this case, 20 generations have been carried out, so that the two cases are compared approximately for the same total number of objective function evaluations; it can be seen how, in the latter case, a much coarser representation of the Pareto front is obtained (43 individuals).

In Fig. 6 the execution times of the parallel genetic algorithm are compared with those of the scalar version. The comparison is made running for 10 generations the same wing planform optimization problem described above. The speed-up obtained is generally very good, being very close to the number of available processors. It should be observed that the scalar version is faster than the parallel one when only one slave processor is activated; this behavior is due to synchronization and context switching overheads.

In Fig. 7 the values of the design parameters of the solutions belonging to the Pareto front are shown as a function of aerodynamic drag. It is interesting to observe how the variation of each design parameter along the front is far from being linear; in particular, "peaks" of some of the design variables can be found in correspondence of "valleys" of others. The distributions are nevertheless piecewise linear, which explains why also a linear interpolation between two elements on the Pareto front is likely to belong, in turn, to the Pareto front.

6.3. Wing section optimization

The design problem here presented consists in the minimization of (inviscid) drag for the ONERA M6 wing, at the design point $M = 0.84$, $c_L = 0.3$. The wing planform shape and the maximum thickness of the wing sections have been kept constant in the optimization process. This problem is first solved without additional constraints; only a geometrical constraint on the minimum allowable trailing edge angle (γ_{TE}) is used to avoid unfeasible geometries. The cases of a twisted and untwisted wing are considered separately. Afterwards, the same problem is faced introducing the requirement of controlling the value of the pitching moment coefficient during the optimization process, and an additional constraint on the leading edge radius to avoid undesirable off design performances. The formulation of these design problems is summarized in Table 3.

6.3.1 Drag minimization

The results obtained for the optimization problems #1 and #2 of Table 2 are here illustrated. The objective function to minimize was c_D/c_L^2 (to account for small variations of the lift coefficient around the design value), with the fitness obtained as $f = 1/obj^2$. The constraint on the maximum thickness of the wing sections, so as to maintain it at the same value of the original geometry, is imposed by scaling the sections to the desired thickness after each geometry modification. Finally, the constraint on the trailing edge angle has been imposed as a filter, by assigning a very high value to the objective function of those geometries which violate it, and skipping the aerodynamic analysis.

The wing has been assigned using 4 spanwise sections, at the positions η =0, 0.33, 0.66 and 1.0, and the optimization has been carried out beginning at the root and progressing to the tip in 4 subsequent cycles: in the first one, the modification to the airfoil shape was maintained constant in the spanwise direction; in the second cycle, the section at $\eta = 0$ was frozen, and the wing was modified from the section at $\eta = 0.33$ to the tip; in the third and fourth cycles, the same was done considering the remaining sections. A total of 12 design variables have been used for the wing sections modification.

The parameters used for the GA are reported in Table 2. As can be seen, selection was carried out using a 2 step random-walk, crossover was the extended intermediate recombination with probability 1, mutation was carried out at word level with probability 0.05, and the population size was 32.

As previously stated the described optimization run was first carried out considering an untwisted wing geometry; afterwards, a second optimization run with the same characteristics has been carried out, adding also the twist angle among the design variables. The twist was supposed to vary linearly from root to tip, so there was one extra design variable for it. Figure 8 illustrates the convergence histories obtained for the untwisted and twisted wings; as can be seen, 10 generations were used for the first modification cycle, and 5 for the subsequent ones. A total number of (approximately) 800 aerodynamic analyses were thus required. In Table 4 the aerodynamic coefficients of the original and modified wings at the design point are reported, including the values obtained at the end of the first optimization cycle; also the value of the twist angle θ is showed in the table. It can be seen that, in both the twisted and untwisted cases, most part of the gain in aerodynamic efficiency is obtained in the first optimization cycle, i.e. for a spanwise constant wing section. For the untwisted geometry the decrease obtained for inviscid drag is $\Delta c_D = 53.3$ d.c., which becomes $\Delta c_D = 58.0$ d.c. when twisting is allowed. This is essentially due to the reduction of wave drag, as induced drag is almost constant. Figure 9 illustrates the drag rise curves (at $c_L = 0.3$) of the ONERA M6 and the two optimized geometries. In Fig. 10 the shape of the airfoils of the control stations are illustrated, while Figures 11 and 12 show the Mach number distribution over the upper side of the original and optimized geometries. From these it can be observed how the suction peaks at the leading edge

are noticeably reduced, and the recompression from supersonic regime is moved close to the trailing edge, resulting in a marked decrease of the pitching moment coefficient. This situation is more apparent when twisting is allowed, especially in the outboard part of the wing; in fact, as can be seen from Fig. 10, the shape of the airfoils is supercritical from root to tip, whereas in the untwisted case the supercritical characteristic is lost when moving towards the tip. A difference between the two cases is also the leading edge radius of the wing sections, which is lower when the wing is twisted as can be expected.

6.3.2 Drag minimization with constrained pitching moment

In the second case considered an additional requirement regarding the pitching moment coefficient has been introduced as shown in Table 3. The pitching moment coefficient of the ONERA M6 wing at the design point is $c_M = -0.1315$ (evaluated using the full-potential flow model); thus, the considered constraint $c_M \geq -0.135$ corresponds to a maximum allowable decrease of about 2.5%. A multiobjective approach has been adopted in this case, with $obj_1 = c_D/c_L^2$, $obj_2 = (c_M - \tilde{c}_M)^2$. The value $\tilde{c}_M = -0.130$ has been used; in this way, the solutions on the final Pareto front will be characterized by pitching moment coefficients in the range from approximately $\tilde{c}_M$ to a lower value corresponding to the minimum drag solutions. Like in the optimization previously described, two separate runs have been carried out by allowing or not the wing twist to be modified; for the sake of simplicity, in this case the shape of the wing sections has been kept constant in the spanwise direction. The same parameters of Table 2 have been used for the GA, except for the population size which has been increased to 48 elements, and the length of the random walk which has been set to three steps. In Fig. 13 the solutions belonging to the final Pareto fronts are illustrated; it must be noted that the solutions are not reported in the objectives plane, but directly in the $c_d - c_M$ plane, which explains why their distribution doesn't look exactly as a Pareto front.

It is now possible to choose a solution with the desired characteristics among those obtained; the aerodynamic coefficients of the minimum drag solutions which satisfy the constraint on pitching moment are reported in Table 5. Fig. 14 shows the drag rise curves at $c_L = 0.3$ of these wings compared to the original wing. The twisted wing, though characterized by lower drag at the design Mach number, shows a higher drag at lower Mach numbers (drag creep) respect to the untwisted wing.

It can be seen that, in the case of the untwisted solution, the decrease in aerodynamic drag is approximately 45 drag counts, with the pitching moment coefficient within the acceptable value. On the other hand, with the twisted solution a decrease in aerodynamic drag of approximately 54 drag counts is obtained, and the pitching moment coefficient is even reduced with respect to the initial value. In Fig.s 15 and 16 the pressure distributions over the original and optimized wings are shown. When the wing is allowed to twist. it is possible to see how the intensity of the shock is reduced, while its position is to some extent anticipated in the outboard portion of the wing. Finally, In Fig. 17 the modified airfoil shapes are illustrated.

As can be seen, considering that the constraint on the pitching moment doesn't allow the optimizer to change the camber of the wing section very much, the final geometries in both cases are characterized by a significant reduction of the leading edge radius (see Table 5). This can be an unacceptable modification, as it may lead to unsatisfactory off design performances, in particular regarding high lift conditions (c_L^{MAX} of the clean wing). For this reason, the optimization procedure as described before has been repeated for the untwisted and twisted cases introducing an additional constraint on the leading edge radius, so that this could not be reduced to values lower than 1% of the local chord. In Fig. 18 the Pareto fronts thus obtained are shown, compared to those obtained without leading edge radius constraint. It may be observed how the solutions on these Pareto fronts are similar to those obtained without the leading edge radius constraint at the low drag end of the fronts, whereas they are characterized by higher drag values in the low pitching moment region of the fronts. Considering the same constraint $c_M > -0.135$, the characteristics of the solutions that can be extracted from these fronts are illustrated in Table 6.

As can be seen, the drag penalty of these solutions with respect to the corresponding ones without constraints on the leading edge radius is approximately 6.9 drag counts for the untwisted case, and 6.5 in the twisted one. In Fig. 19 the drag rise curves at the design lift coefficient of the optimized wings with and without the constraint on leading edge radius are illustrated.

Fig.s 20 and 21 illustrate the pressure distributions over the original and optimized wings, whereas the corresponding geometries are illustrated in Fig. 22. When the wing is twisted, the position of the shock wave is slightly anticipated, but its intensity is reduced.

7. Conclusions

A multiobjective genetic algorithm has been used for the optimization of a wing in transonic flow. In the design problems that have been described both geometrical and aerodynamic constraints have been taken into consideration. It has been shown in particular how multiobjective optimization can be an effective approach when conflicting design criteria must be met; in the application described this approach has been used to devise an optimum planform shape taking into account both aerodynamic drag and structural weight, and to control the value of the pitching moment while minimizing aerodynamic drag. A full-potential flow solver has been used as analysis tool; comparison of the results obtained with this flow solver with the corresponding ones obtained through Euler and Navier-Stokes solvers show that the full-potential model may be considered adequate for transonic cruise conditions, when viscous effects are not very important[14]. Simple semi-empirical criteria, like constraints on the maximum allowable Mach number ahead of shocks to avoid flow separation, may then be sufficient. On the other hand, if design problems of industrial relevance need to be faced, it is necessary to take into account complex geometries, and also viscous effects are to be included directly in the design loop to obtain more reliable solutions. Though genetic algorithms are effective and robust design tools, well suited for multidisciplinary environments, the critical issue still lies in

the computational effort they require. This makes their application unpractical when the fitness evaluation becomes computationally expensive. Two possible approaches to the problem have been investigated in this work, consisting in the hybridization of the algorithm to exploit the favourable features of gradient based search methods, and in the use of parallel computing.

References

[1]Lynch, F.T., "Commercial Transports - Aerodynamic Design for Cruise Performance Efficiency," Transonic Perspective Symposium, Progress in Aeronautics and Astronautics, Vol.81, D. Nixon ed., AIAA, New York, 1982, pp.81-147.

[2] *Optimum Design Methods for Aerodynamics*, AGARD-R-803, Nov. 1994.

[3]Hajela, P., "Nongradient Methods in Multidisciplinary Design Optimization - Status and Potential," J. of Aircraft, Vol.36, No.1, Jan.-Feb. 1999, pp. 255-265.

[4]Vicini, A., Quagliarella, D., "Multipoint Transonic Airfoil Design by Means of a Multiobjective Genetic Algorithm," AIAA Paper 97-0082, Jan. 1997.

[5]Holland, J. H., *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Michigan, 1975.

[6]Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Massachusetts, Jan. 1989.

[7]Vicini, A., Quagliarella, D., "Inverse and Direct Airfoil Design Using a Multiobjective Genetic Algorithm," *AIAA Journal*, Vol. 35, No. 9, Sep. 1997, pp. 1499–1505.

[8] *CAST-10-2/DOA 2, Airfoil Studies Workshop Results*, NASA CP 3052, 1989.

[9]Quagliarella, D., Vicini, A., "Subpopulation policies for a parallel multiobjective genetic algorithm with applications to wing design," IEEE SMC'98 Congress, San Diego, October 1998.

[10]D. Cortesi, A. Evans, W. Ferguson, J. Hartman, *Topics in IRIX™ Programming — Chapter 8, Models of Parallel Computation*, Silicon Graphicsi Inc., 1996, Document Number 007-2478-004.

[11]Schmitt, V., Charpin, F., "Pressure distributions on the ONERA M6 wing at transonic Mach numbers," in *Experimental data base for computer program assessment*, AGARD-AR-138, Paper B1, May 1979.

[12]Torenbeek, E., "Development and Application of a Comprehensive, Design-Sensitive Weight Prediction Method for Wing Structures of Transport Category Aircraft," Technical Report LR-693, TU Delft, Sep. 1992.

[13]A. Vicini, D. Quagliarella, "Airfoil and Wing Design Through Hybrid Optimization Strategies," *AIAA Journal*, Vol. 37, No. 5, May 1997.

[14]A.Vicini "Transonic wing design using genetic algorithms", Report CIRA TR-98-212, December 1998.

design variable	initial value	allowable range
λ	0.562	[0.2 , 0.8]
Λ	30.0	[15 , 36]
θ	0.0	[-10 , 10]
AR	3.8	[3.5 , 4.2]
$t/c\|_r$ %	9.8	[8 , 14]

Table 1 - Design parameters for the wing planform optimization.

	Planform optimization	Section optimization
Selection	Random-Walk, 2 steps	Random-Walk, 2 steps
Crossover	One-point	Extended Intermediate Recombination
P_c	1	1
Mutation	Bit	Word
P_m	0.1	0.05
Pop size	16	32

Table 2 - Parameters for the GA.

Design problem #	1	2	3	4	5	6
Mach	0.84					
c_L	0.3					
c_D	minimize					
c_M	free		$\geq$-0.135			
t/c%	fixed = 9.8%					
twist	no	yes	no	yes	no	yes
γ_{TE}	$\geq 10^\circ$					
L.E. radius	free				$\geq$ 1% c	

Table 3 - Formulation of the design problems for the M6 section optimization.

Wing #	α	c_L	c_D	c_{Di}	c_{Dw}	c_M	θ
M6	3.3167	0.3000	0.01369	0.00700	0.00670	-0.13151	0.
1	1.6271	0.3002	0.00887	0.00705	0.00182	-0.22807	0.
2	1.4482	0.3004	0.00836	0.00711	0.00125	-0.23357	0.
3	1.1830	0.3007	0.00843	0.00706	0.00136	-0.22465	-1.4°
4	0.8530	0.3028	0.00789	0.00722	0.00067	-0.24006	-3.5°

Table 4 - Aerodynamic coefficients of the optimized wings (design problems 1,2). Wing 1: untwisted, constant airfoil; wing 2: untwisted, variable airfoil; wing 3: twisted, constant airfoil; wing 4: twisted, variable airfoil.

Wing #	α	c_L	c_D	c_{Di}	c_{Dw}	c_M	θ	r_{LE}
M6	3.3167	0.3000	0.01369	0.00700	0.00670	-0.13151	0.0	1.50%
1	3.4558	0.3000	0.00918	0.00700	0.00219	-0.13476	0.0	0.88%
2	2.8994	0.3002	0.00830	0.00731	0.00099	-0.12970	-4.09	0.37%

Table 5 - Aerodynamic coefficients of the optimized wings (design problems 3,4). Wing 1: untwisted; wing 2: twisted.

Wing #	α	c_L	c_D	c_{Di}	c_{Dw}	c_M	θ	r_{LE}
M6	3.3167	0.3000	0.01369	0.00700	0.00670	-0.13151	0.00	1.50%
1	3.3823	0.3000	0.00987	0.00700	0.00288	-0.13475	0.00	1.14%
2	2.6742	0.3000	0.00895	0.00748	0.00147	-0.13237	-5.88	1.03%

Table 6 - Aerodynamic coefficients of the optimized wings (design problems 5,6). Wing 1: untwisted; wing 2: twisted.

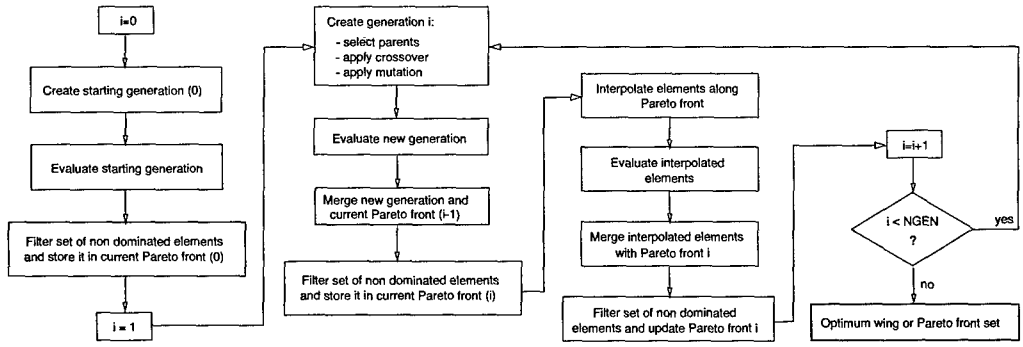

Fig. 1 - Flow chart of the multiobjective GA

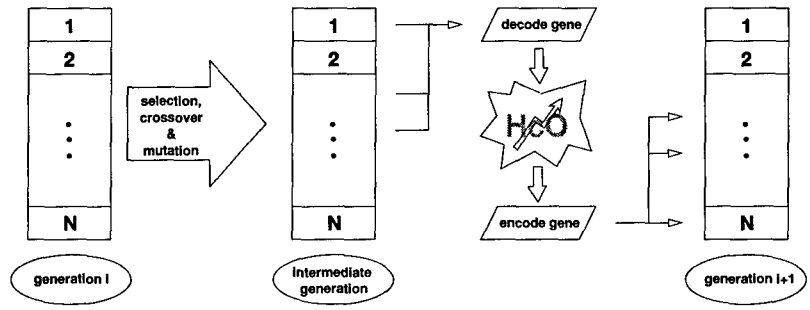

Fig. 2 - Sketch of the hybrid GA

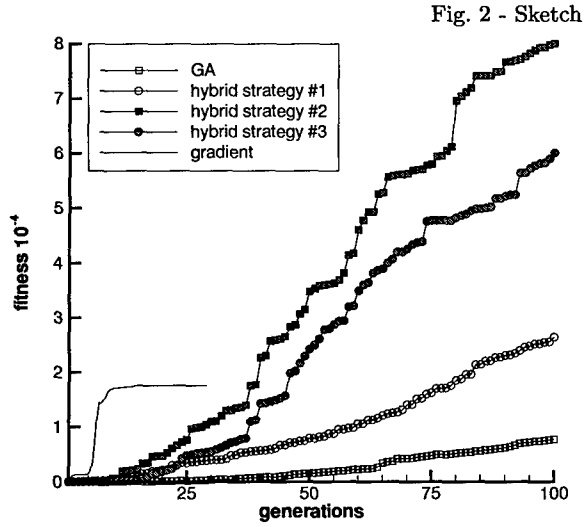

Fig. 3 - Convergence histories of the inverse airfoil design problem (averaged over 10 trials)

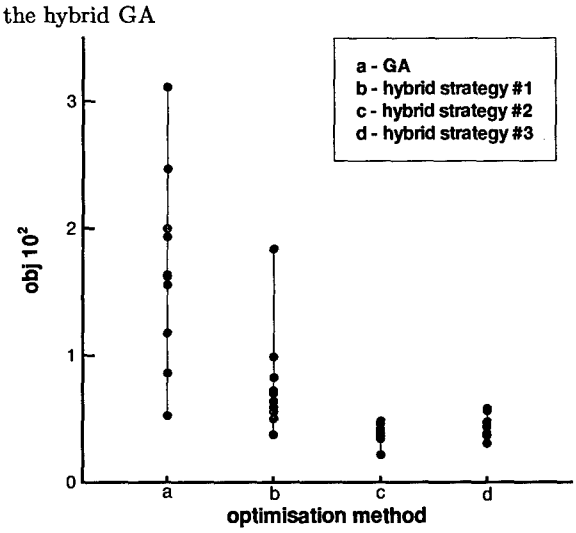

Fig. 4 - Scatter of results obtained in 10 different runs for the inverse airfoil design problem

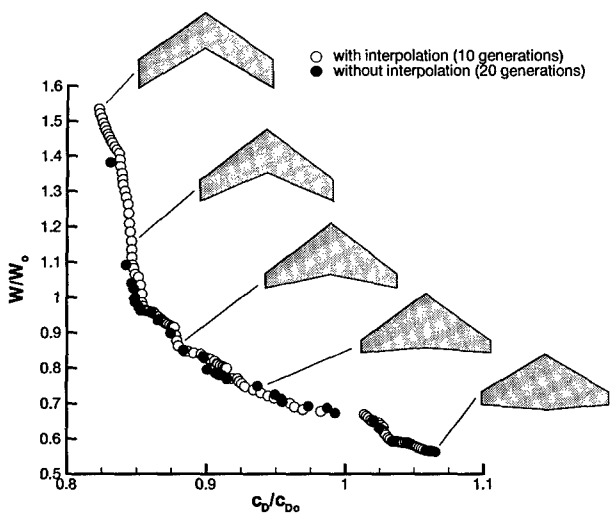

Fig. 5 - Pareto fronts for the planform optimization

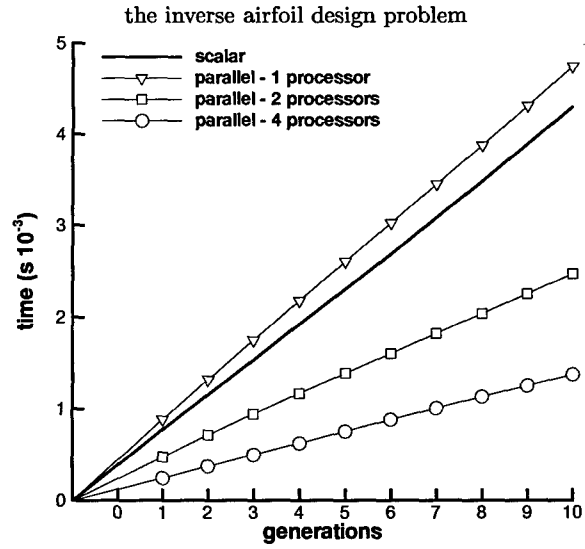

Fig. 6 - CPU times for the scalar and parallel GA

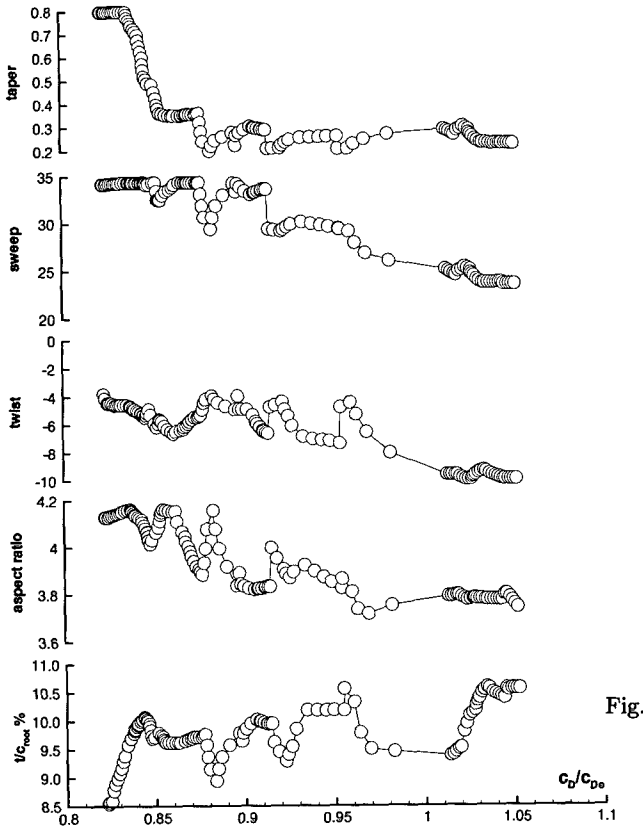

Fig. 7 - Parameters of the solutions on the Pareto front

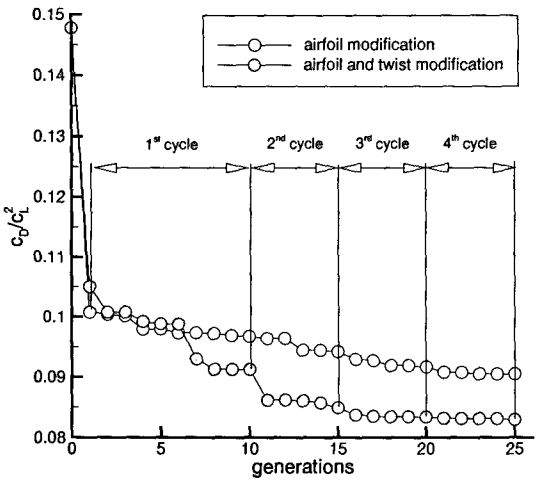

Fig. 8 - Convergence histories for the drag minimization problem

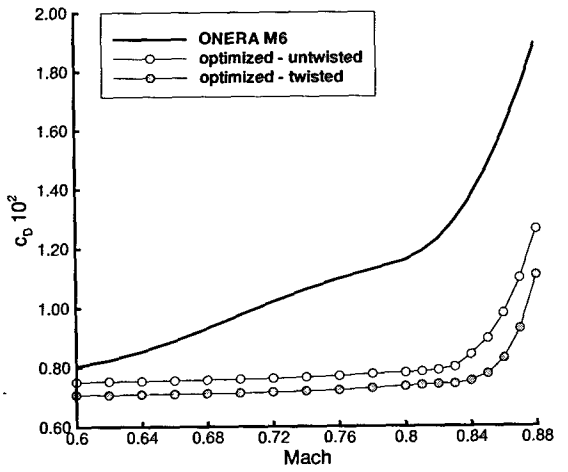

Fig. 9 - Drag rise curves for the original and optimized wings

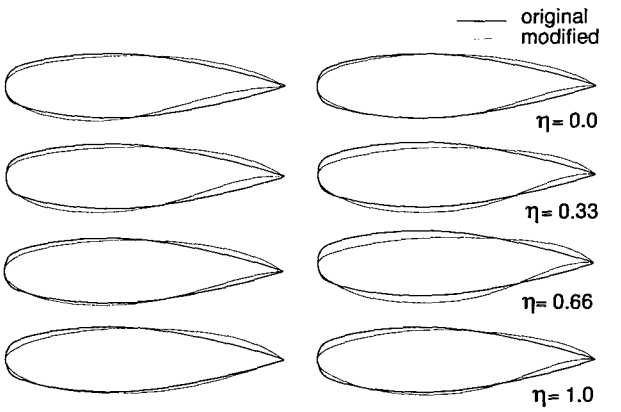

Fig. 10 - Shape of the modified airfoils; untwisted (left) and twisted (right) wings

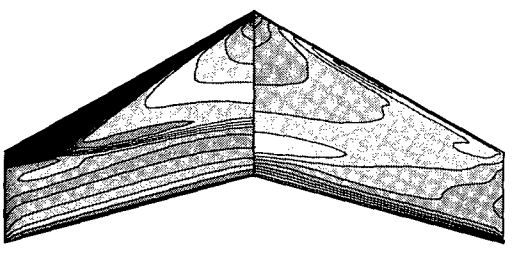

Fig. 11 - Mach number distribution on the original (left) and optimized (untwisted) wing

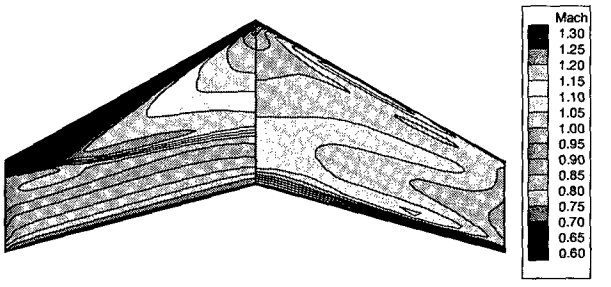

Fig. 12 - Mach number distribution on the original (left) and optimized (twisted) wing

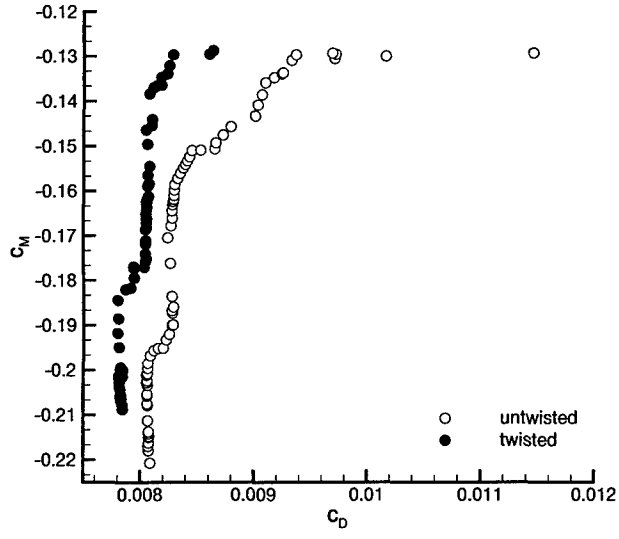

Fig. 13 - Pareto fronts obtained

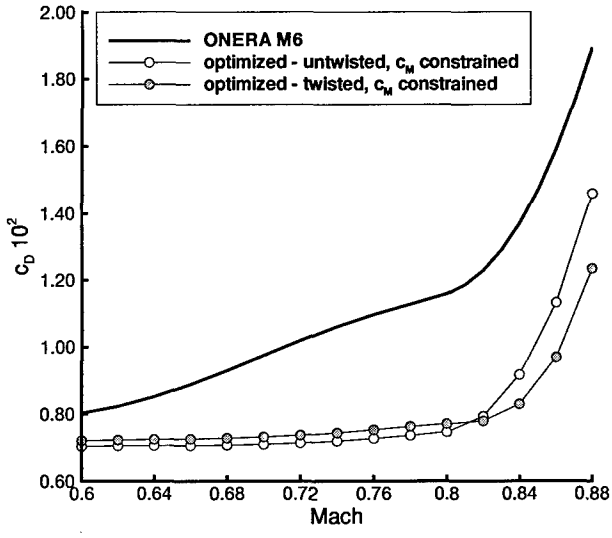

Fig. 14 - Drag rise curves for the original and optimized wings

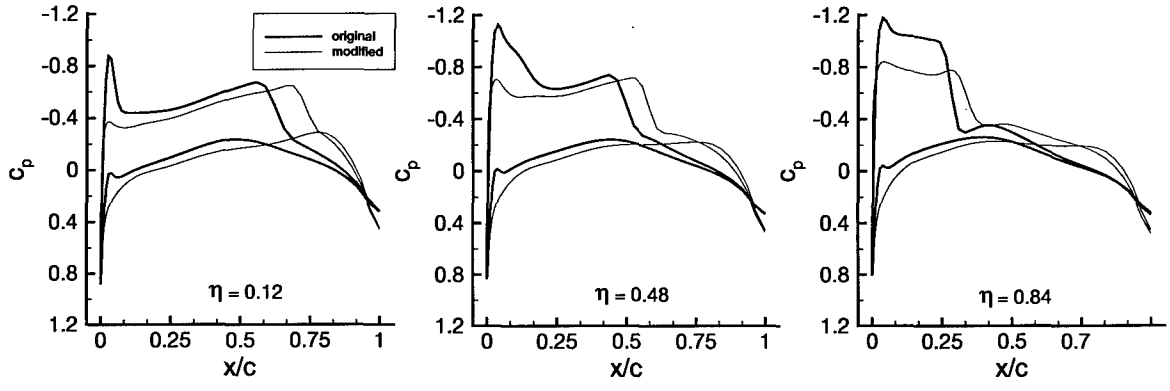

Fig. 15 - Pressure coefficient distributions on the original and optimized (untwisted) wings

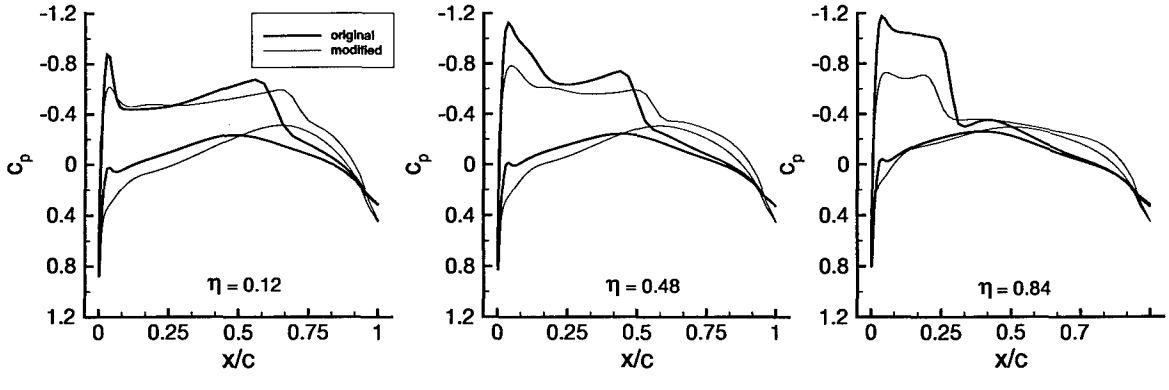

Fig. 16 - Pressure coefficient distributions on the original and optimized (twisted) wings

Fig. 17 - Shape of the modified airfoils for the untwisted (left) and twisted (right) wings

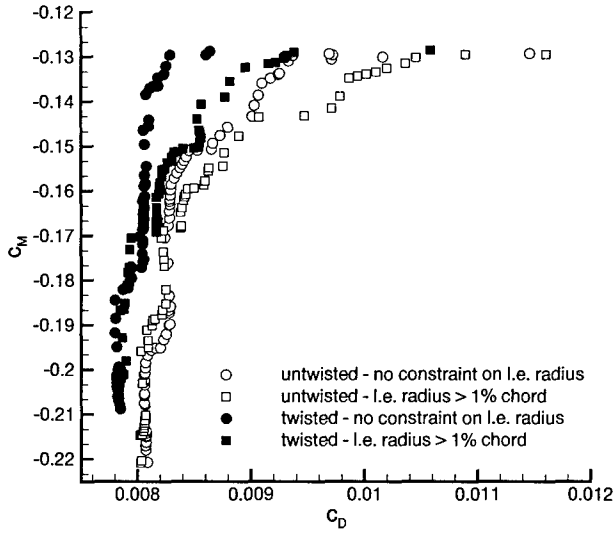

Fig. 18 - Pareto fronts obtained with and without leading edge radius constraint

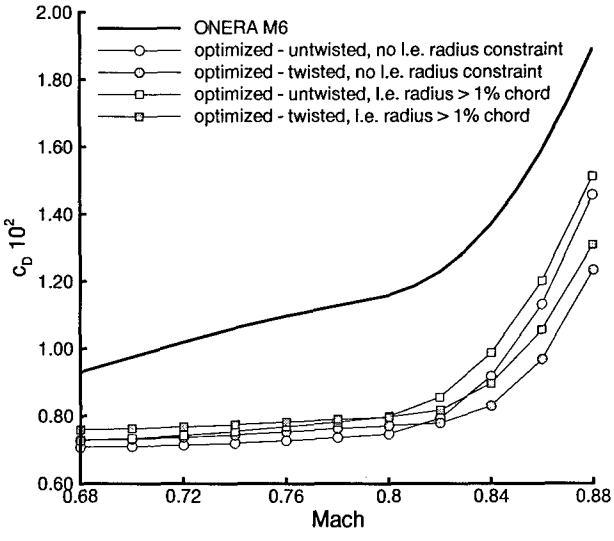

Fig. 19 - Drag rise curves for the original and optimized wings with and without leading edge radius constraint

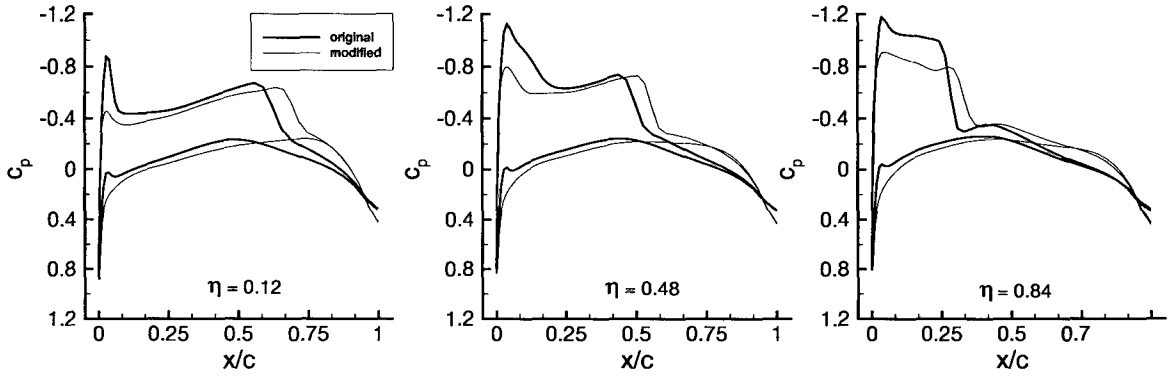

Fig. 20 - Pressure coefficient distributions on the original and optimized (untwisted) wings, with l.e. radius constraint

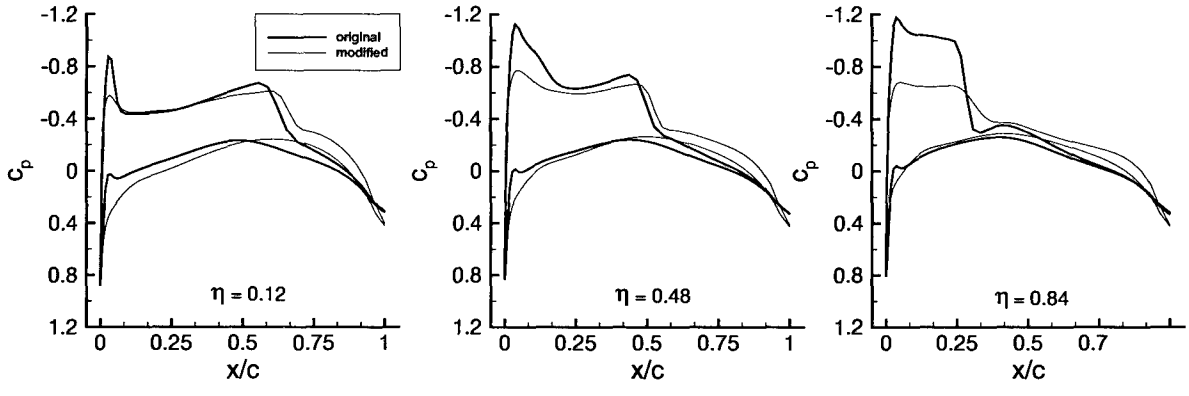

Fig. 21 - Pressure coefficient distributions on the original and optimized (twisted) wings, with l.e. radius constraint

Fig. 22 - Shape of the modified airfoils for the untwisted (left) and twisted (right) wing, with l.e. radius constraint